Source code:

**package** com.ooyala.android;

**import** com.ooyala.android.player.PlayerInterface;

**import** com.ooyala.android.plugin.AdPluginInterface;

**import** com.ooyala.android.plugin.LifeCycleInterface;

**import** com.ooyala.android.util.DebugMode;

**import** java.util.ArrayList;

**import** java.util.HashSet;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Set;

**class** AdPluginManager

**implements** AdPluginManagerInterface, LifeCycleInterface

{

**private** **static** **final** String TAG = AdPluginManager.**class**.getName();

**private** AdPluginInterface \_activePlugin = **null**;

**private** AdPluginManagerInterface.AdMode \_admode = AdPluginManagerInterface.AdMode.None;

**private** **int** \_parameter = 0;

**private** OoyalaPlayer \_player;

**private** List<AdPluginInterface> \_plugins = **new** ArrayList();

**public** AdPluginManager(OoyalaPlayer paramOoyalaPlayer)

{

**this**.\_player = paramOoyalaPlayer;

}

**private** **static** AdPluginInterface getNextPlugin(List<AdPluginInterface> paramList, AdPluginInterface paramAdPluginInterface)

{

**if** (paramList.size() == 0) {}

**int** i;

**do**

{

**return** **null**;

**if** (paramAdPluginInterface == **null**) {

**return** (AdPluginInterface)paramList.get(0);

}

DebugMode.assertCondition(paramList.contains(paramAdPluginInterface), TAG, "the list does not contain plugin " + paramAdPluginInterface.toString());

i = paramList.indexOf(paramAdPluginInterface);

} **while** ((i < 0) || (i >= paramList.size() - 1));

**return** (AdPluginInterface)paramList.get(i + 1);

}

**private** **boolean** pluginNeedsAdMode(AdPluginInterface paramAdPluginInterface, AdPluginManagerInterface.AdMode paramAdMode)

{

**if** (paramAdPluginInterface == **null**)

{

DebugMode.assertFail(TAG, "plugin method is called when active plugin is null");

**return** **false**;

}

**switch** (paramAdMode)

{

**default**:

DebugMode.assertFail(TAG, "request admode when admode is not defined");

**return** **false**;

**case** ???:

**return** paramAdPluginInterface.onContentChanged();

**case** ???:

**return** paramAdPluginInterface.onInitialPlay();

**case** ???:

**return** paramAdPluginInterface.onPlayheadUpdate(**this**.\_parameter);

**case** ???:

**return** paramAdPluginInterface.onCuePoint(**this**.\_parameter);

**case** ???:

**return** paramAdPluginInterface.onContentFinished();

}

**return** paramAdPluginInterface.onContentError(**this**.\_parameter);

}

**public** AdPluginManagerInterface.AdMode adMode()

{

**return** **this**.\_admode;

}

**public** **boolean** deregisterPlugin(AdPluginInterface paramAdPluginInterface)

{

**if** (!**this**.\_plugins.contains(paramAdPluginInterface))

{

DebugMode.logD(TAG, paramAdPluginInterface.toString() + "is not registered or has been removed");

**return** **false**;

}

**if** (**this**.\_activePlugin == paramAdPluginInterface)

{

DebugMode.assertFail(TAG, "try to deregister when the plugin is still active");

**return** **false**;

}

**this**.\_plugins.remove(paramAdPluginInterface);

DebugMode.logD(TAG, "deregister ad plugin" + paramAdPluginInterface.toString());

**return** **true**;

}

**public** **void** destroy()

{

**if** (**this**.\_activePlugin != **null**) {

**this**.\_activePlugin.destroy();

}

}

**public** **boolean** exitAdMode(AdPluginInterface paramAdPluginInterface)

{

**if** (paramAdPluginInterface == **null**)

{

DebugMode.assertFail(TAG, "exitAdModed.plugin is null");

**return** **false**;

}

**if** (!**this**.\_plugins.contains(paramAdPluginInterface))

{

DebugMode.assertFail(TAG, paramAdPluginInterface.toString() + " exit admode before it register");

**return** **false**;

}

**if** (**this**.\_activePlugin != paramAdPluginInterface)

{

**if** (**this**.\_activePlugin != **null**)

{

DebugMode.assertFail(TAG, paramAdPluginInterface.toString() + " exit admode but active plugin is " + **this**.\_activePlugin.toString());

**return** **false**;

}

**return** **true**;

}

AdPluginInterface localAdPluginInterface = **null**;

**if** (**this**.\_admode != AdPluginManagerInterface.AdMode.PluginInitiated) {

**for** (paramAdPluginInterface = getNextPlugin(**this**.\_plugins, paramAdPluginInterface);; paramAdPluginInterface = getNextPlugin(**this**.\_plugins, paramAdPluginInterface))

{

localAdPluginInterface = paramAdPluginInterface;

**if** (paramAdPluginInterface == **null**) {

**break**;

}

localAdPluginInterface = paramAdPluginInterface;

**if** (pluginNeedsAdMode(paramAdPluginInterface, **this**.\_admode)) {

**break**;

}

}

}

**if** (localAdPluginInterface == **null**)

{

paramAdPluginInterface = **this**.\_admode;

**this**.\_admode = AdPluginManagerInterface.AdMode.None;

setActivePlugin(**null**);

**this**.\_player.processExitAdModes(paramAdPluginInterface, **true**);

}

**for** (;;)

{

**return** **true**;

setActivePlugin(localAdPluginInterface);

**this**.\_activePlugin.onAdModeEntered();

}

}

**void** forceExitAdMode()

{

DebugMode.logD(TAG, "forceExitAdMode");

**this**.\_admode = AdPluginManagerInterface.AdMode.None;

**if** ((**this**.\_activePlugin != **null**) && (**this**.\_activePlugin.getPlayerInterface() != **null**)) {

**this**.\_activePlugin.getPlayerInterface().stop();

}

**this**.\_activePlugin = **null**;

}

**public** AdPluginInterface getActivePlugin()

{

**return** **this**.\_activePlugin;

}

**public** Set<Integer> getCuePointsInMilliSeconds()

{

HashSet localHashSet = **new** HashSet();

Iterator localIterator = **this**.\_plugins.iterator();

**while** (localIterator.hasNext()) {

localHashSet.addAll(((AdPluginInterface)localIterator.next()).getCuePointsInMilliSeconds());

}

**return** localHashSet;

}

**public** PlayerInterface getPlayerInterface()

{

**if** (**this**.\_activePlugin != **null**) {

**return** **this**.\_activePlugin.getPlayerInterface();

}

**return** **null**;

}

**public** **boolean** inAdMode()

{

**return** **this**.\_activePlugin != **null**;

}

**boolean** onAdMode(AdPluginManagerInterface.AdMode paramAdMode, **int** paramInt)

{

**this**.\_parameter = paramInt;

**if** (**this**.\_plugins.size() <= 0) {}

AdPluginInterface localAdPluginInterface;

**do**

{

**return** **false**;

**for** (localAdPluginInterface = (AdPluginInterface)**this**.\_plugins.get(0); (localAdPluginInterface != **null**) && (!pluginNeedsAdMode(localAdPluginInterface, paramAdMode)); localAdPluginInterface = getNextPlugin(**this**.\_plugins, localAdPluginInterface)) {}

} **while** (localAdPluginInterface == **null**);

setActivePlugin(localAdPluginInterface);

**this**.\_admode = paramAdMode;

**return** **true**;

}

**public** **void** onAdModeEntered()

{

**if** (**this**.\_activePlugin == **null**)

{

DebugMode.assertFail(TAG, "enter ad mode when active plugin is null");

**return**;

}

**this**.\_activePlugin.onAdModeEntered();

}

**public** **boolean** registerPlugin(AdPluginInterface paramAdPluginInterface)

{

**if** (**this**.\_plugins.contains(paramAdPluginInterface))

{

DebugMode.logD(TAG, "plugin " + paramAdPluginInterface.toString() + "already exist");

**return** **false**;

}

Iterator localIterator = **this**.\_plugins.iterator();

**while** (localIterator.hasNext())

{

AdPluginInterface localAdPluginInterface = (AdPluginInterface)localIterator.next();

**if** (paramAdPluginInterface.getClass() == localAdPluginInterface.getClass()) {

DebugMode.logD(TAG, "plugin " + localAdPluginInterface.toString() + " is same class as " + paramAdPluginInterface.toString());

}

}

DebugMode.logD(TAG, "register ad plugin" + paramAdPluginInterface.toString());

**this**.\_plugins.add(paramAdPluginInterface);

**return** **true**;

}

**public** **boolean** requestAdMode(AdPluginInterface paramAdPluginInterface)

{

**if** (**this**.\_activePlugin != **null**) {

**return** **false**;

}

**this**.\_activePlugin = paramAdPluginInterface;

**this**.\_admode = AdPluginManagerInterface.AdMode.PluginInitiated;

**return** **true**;

}

**public** **void** reset()

{

**if** (**this**.\_activePlugin != **null**) {

**this**.\_activePlugin.reset();

}

}

**public** **void** resetAds()

{

Iterator localIterator = **this**.\_plugins.iterator();

**while** (localIterator.hasNext()) {

((AdPluginInterface)localIterator.next()).resetAds();

}

}

**public** **void** resetManager()

{

**if** (**this**.\_activePlugin != **null**)

{

**this**.\_activePlugin.destroy();

**this**.\_activePlugin = **null**;

}

}

**public** **void** resume()

{

**if** (**this**.\_activePlugin != **null**) {

**this**.\_activePlugin.resume();

}

}

**public** **void** resume(**int** paramInt, OoyalaPlayer.State paramState)

{

**if** (**this**.\_activePlugin != **null**) {

**this**.\_activePlugin.resume(paramInt, paramState);

}

}

**protected** **void** setActivePlugin(AdPluginInterface paramAdPluginInterface)

{

**this**.\_activePlugin = paramAdPluginInterface;

}

**public** **void** skipAd()

{

Iterator localIterator = **this**.\_plugins.iterator();

**while** (localIterator.hasNext()) {

((AdPluginInterface)localIterator.next()).skipAd();

}

}

**public** **void** suspend()

{

**if** (**this**.\_activePlugin != **null**) {

**this**.\_activePlugin.suspend();

}

}

}

CONFIGURATION

**package** com.ooyala.android.configuration;

**public** **class** Options

**implements** ReadonlyOptionsInterface

{

**private** **final** **int** connectionTimeoutInMillisecond;

**private** **final** **boolean** preloadContent;

**private** **final** **boolean** preventVideoViewSharing;

**private** **final** **int** readTimeoutInMillisecond;

**private** **final** **boolean** showAdsControls;

**private** **final** **boolean** showCuePoints;

**private** **final** **boolean** showLiveControls;

**private** **final** **boolean** showPromoImage;

**private** **final** FCCTVRatingConfiguration tvRatingConfiguration;

**private** **final** VisualOnConfiguration visualOnConfiguration;

**private** Options(FCCTVRatingConfiguration paramFCCTVRatingConfiguration, VisualOnConfiguration paramVisualOnConfiguration, **boolean** paramBoolean1, **boolean** paramBoolean2, **boolean** paramBoolean3, **boolean** paramBoolean4, **boolean** paramBoolean5, **int** paramInt1, **int** paramInt2, **boolean** paramBoolean6)

{

**this**.tvRatingConfiguration = paramFCCTVRatingConfiguration;

**this**.visualOnConfiguration = paramVisualOnConfiguration;

**this**.showCuePoints = paramBoolean1;

**this**.showAdsControls = paramBoolean2;

**this**.preloadContent = paramBoolean3;

**this**.showPromoImage = paramBoolean4;

**this**.showLiveControls = paramBoolean5;

**this**.connectionTimeoutInMillisecond = paramInt1;

**this**.readTimeoutInMillisecond = paramInt2;

**this**.preventVideoViewSharing = paramBoolean6;

}

**public** **int** getConnectionTimeoutInMillisecond()

{

**return** **this**.connectionTimeoutInMillisecond;

}

**public** **boolean** getPreloadContent()

{

**return** **this**.preloadContent;

}

**public** **boolean** getPreventVideoViewSharing()

{

**return** **this**.preventVideoViewSharing;

}

**public** **int** getReadTimeoutInMillisecond()

{

**return** **this**.readTimeoutInMillisecond;

}

**public** **boolean** getShowAdsControls()

{

**return** **this**.showAdsControls;

}

**public** **boolean** getShowCuePoints()

{

**return** **this**.showCuePoints;

}

**public** **boolean** getShowLiveControls()

{

**return** **this**.showLiveControls;

}

**public** **boolean** getShowPromoImage()

{

**return** **this**.showPromoImage;

}

**public** FCCTVRatingConfiguration getTVRatingConfiguration()

{

**return** **this**.tvRatingConfiguration;

}

**public** VisualOnConfiguration getVisualOnConfiguration()

{

**return** **this**.visualOnConfiguration;

}

**public** **static** **class** Builder

{

**private** **int** connectionTimeoutInMillisecond = 0;

**private** **boolean** preloadContent = **true**;

**private** **boolean** preventVideoViewSharing = **false**;

**private** **int** readTimeoutInMillisecond = 0;

**private** **boolean** showAdsControls = **true**;

**private** **boolean** showCuePoints = **true**;

**private** **boolean** showLiveControls = **true**;

**private** **boolean** showPromoImage = **false**;

**private** FCCTVRatingConfiguration tvRatingConfiguration = FCCTVRatingConfiguration.s\_getDefaultTVRatingConfiguration();

**private** VisualOnConfiguration visualOnConfiguration = VisualOnConfiguration.s\_getDefaultVisualOnConfiguration();

**public** Options build()

{

**return** **new** Options(**this**.tvRatingConfiguration, **this**.visualOnConfiguration, **this**.showCuePoints, **this**.showAdsControls, **this**.preloadContent, **this**.showPromoImage, **this**.showLiveControls, **this**.connectionTimeoutInMillisecond, **this**.readTimeoutInMillisecond, **this**.preventVideoViewSharing, **null**);

}

**public** Builder setConnectionTimeout(**int** paramInt)

{

**this**.connectionTimeoutInMillisecond = paramInt;

**return** **this**;

}

**public** Builder setPreloadContent(**boolean** paramBoolean)

{

**this**.preloadContent = paramBoolean;

**return** **this**;

}

**public** Builder setPreventVideoViewSharing(**boolean** paramBoolean)

{

**this**.preventVideoViewSharing = paramBoolean;

**return** **this**;

}

**public** Builder setReadTimeout(**int** paramInt)

{

**this**.readTimeoutInMillisecond = paramInt;

**return** **this**;

}

**public** Builder setShowAdsControls(**boolean** paramBoolean)

{

**this**.showAdsControls = paramBoolean;

**return** **this**;

}

**public** Builder setShowCuePoints(**boolean** paramBoolean)

{

**this**.showCuePoints = paramBoolean;

**return** **this**;

}

**public** Builder setShowLiveControls(**boolean** paramBoolean)

{

**this**.showLiveControls = paramBoolean;

**return** **this**;

}

**public** Builder setShowPromoImage(**boolean** paramBoolean)

{

**this**.showPromoImage = paramBoolean;

**return** **this**;

}

**public** Builder setTVRatingConfiguration(FCCTVRatingConfiguration paramFCCTVRatingConfiguration)

{

**this**.tvRatingConfiguration = paramFCCTVRatingConfiguration;

**return** **this**;

}

**public** Builder setVisualOnConfiguration(VisualOnConfiguration paramVisualOnConfiguration)

{

**this**.visualOnConfiguration = paramVisualOnConfiguration;

**return** **this**;

}

}

}